International Standard

ISO/IEC 23008-12

Information technology — High efficiency coding and media delivery in heterogeneous environments —

Part 12: Image File Format

AMENDMENT 2: Low-overhead image file format

Technologies de l'information — Codage à haute efficacité et livraison des medias dans des environnements hétérogènes —

Partie 12: Format de fichier d'image

AMENDEMENT 2: Format de fichier image à entête optimisé

Third edition
2025-07

AMENDMENT 2
2026-06

Reference number
ISO/IEC 23008-12:2025/Amd. 2:2026(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 23008 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Information technology — High efficiency coding and media delivery in heterogeneous environments —

Part 12: Image File Format

AMENDMENT 2: Low-overhead image file format

Introduction

Add the following paragraphs at the end of the Introduction:

Annex O specifies the Low-overhead Image File Format suitable for use cases requiring small and simple images. This format reduces the overhead relative to the structure-data describing the image or metadata payloads for small image files compared to the regular Image File Format. Annex O also specifies the requirements and brands for this format, as well as the procedure for expanding it to the regular Image File Format.

Annex P specifies the MIME type registration for a single image for the Low-overhead Image File Format-specific brand.

3.1.11

Add the following note to entry after the definition:

Note 1 to entry The term “track” is defined in ISO/IEC 14496-12.

3.1.58

Add the following new term and definition:

3.1.58 tiled image item

image item of type `'tili'` constructed using uniform tiled subregions, allowing direct access to individual tiles

6.1

Add the following NOTE after the first paragraph*:*

NOTE For small and simple files, a pre-processed version of the `MetaBox` called the `MinimizedImageBox`, as defined in Annex O, can be used. A file containing a `MinimizedImageBox` has the same compliance and rendering requirements as an Image File Format file because the `MinimizedImageBox` is expanded into the equivalent `MetaBox` and `MediaDataBox` as described in detail in Annex O.4. Once expanded, the rest of Clause 6 applies.

6.5.1

Number the existing NOTE as NOTE 2.

Replace the third and fourth paragraphs with:

The semantics of the descriptive properties specified in 6.5 are specified for the image before the transformations, if any, are applied.

NOTE 1 It is uncertain if readers would be able to correctly interpret descriptive properties that follow the first transformative property or the first unrecognized essential property, whichever is earlier, in the sequence associating properties with an item, because those descriptive properties possibly describe the output image after the transformation(s). In previous versions of this document, readers had to allow and ignore descriptive properties following the first transformative or unrecognized property, whichever is earlier, in the sequence associating properties with an item.

Writers should arrange the descriptive properties prior to any transformative property in the sequence associating properties with an item. Writers should arrange the descriptive properties specified in 6.5 prior to any other properties in the sequence associating properties with an item.

6.5.6.1

Replace:

The `PixelInformationProperty` descriptive item property indicates the number and bit depth of colour components in the reconstructed image of the associated image item.

with:

The `PixelInformationProperty` descriptive item property indicates the number and bit depth of colour and alpha/depth components, if present, in the reconstructed image of the associated image item. If `px_flags` & 1 != 0, the `PixelInformationProperty` also indicates content, component format and subsampling information per channel.

6.5.6.2

Replace the existing text with the following:

```
aligned(8) class PixelInformationProperty
   extends ItemFullProperty('pixi', version = 0, px_flags){
   unsigned int(8) num_channels;
   for (i=0; i<num_channels; i++) {
      unsigned int(8) bits_per_channel;
   }
   if((px_flags & 1) != 0) {
      for (i=0; i<num_channels; i++) {
         unsigned int(3) channel_idc;
         unsigned int(1) reserved = 0;
         unsigned int(2) component_format;
         unsigned int(1) subsampling_flag;
         unsigned int(1) channel_label_flag;
         if(subsampling_flag) {
            unsigned int(4) subsampling_type;
            unsigned int(4) subsampling_location;
         }
         if(channel_label_flag) {
            utf8string channel_label;
         }
      }
   }
}
```

6.5.6.3

Replace:

`bits_per_channel`: This field indicates the bits per channel for the pixels of the reconstructed image of the associated image item.

With:

`bits_per_channel`: This field indicates the bits per channel for the pixels of the reconstructed image of the associated image item. The value of this field shall not be 0.

6.5.6.3

Add the following text to the end of the subclause:

`px_flags&1`: If equal to 1, indicates that the `channel_idc`, `component_format`, `subsampling_flag`, and `channel_label_flag` fields are present. If equal to 0, indicates that the `channel_idc`, `component_format`, `subsampling_flag` and `channel_label_flag` fields are not present.

`channel_idc`: This field indicates the contents of the channel as specified in Table 13. At most one channel shall have a `channel_idc` of 5.

Table 13 — `channel_idc` values and their meaning

Value of `channel_idc`	Mapping (depending on the `'colr'` box)
0	Unused
1	Unspecified
2	First colour channel (e.g. monochrome, Y, R, C)
3	Second colour channel (e.g. U, Cb, G, M)
4	Third colour channel (e.g. V, Cr, B, Y)
5	Alpha
6	Depth
7	Fourth colour channel (e.g. K)

`component_format`: This field indicates the data type of the channel as defined by the `component_format` values in ISO/IEC 23001-17 where `component_bit_depth` is considered to be equal to `bits_per_channel`.

`subsampling_flag`: If equal to 1, indicates that the `subsampling_type` and `subsampling_location` fields are present. If equal to 0, indicates that the `subsampling_type` and `subsampling_location` fields are not present.

`channel_label_flag`: If equal to 1, indicates the presence of the `channel_label` field. If equal to 0, indicates the `channel_label` field is not present.

`subsampling_type`: This field indicates the subsampling type as specified by Table 14.

`subsampling_location`: This field indicates the subsampling sample location as specified by Table 14.

`channel_label`: The human readable description of the channel.

Table 14 — Channel subsampling and sample position

Value of `subsampling_type`	Value of `subsampling_location`	Channel subsampling	Position of the centre of the top-left chroma sample relative to the centre of the top-left luma sample in units of luma samples	
			Horizontal (x)	Vertical (y)
0	0 or 1 or 2 or 3 or 4 or 5	none (4:4:4)	0	0
1	0 or 2 or 4	subsampled by a factor 2 horizontally (4:2:2)	0	0
	1 or 3 or 5		0.5	
2	0	subsampled both horizontally and vertically by a factor 2 (4:2:0)	0	0.5
	1		0.5	0.5
	2		0	0
	3		0.5	0
	4		0	1
	5		0.5	1
3	0 or 2 or 4	subsampled by a factor 4 horizontally (4:1:1)	0	0
	1 or 3 or 5		1.5	
4	0 or 1	subsampled by a factor 2 vertically (4:4:0)	0	0.5
	2 or 3			0
	4 or 5			1
Other values		Reserved		

6.5.8.2

Change the syntax to:


```
aligned(8) class AuxiliaryTypeProperty
extends ItemFullProperty('auxC', version = 0, flags) {
   utf8string aux_type;
   unsigned int(8) aux_subtype[];
      // until the end of the box, the semantics depend on the aux_type value
}
```

6.5.10.3, 6.5.13.1, 6.6.2.3.1

Replace all occurrences of "*" with "×".

6.6.2.3.1

Set all occurrences of "tile_width" and "tile_height" in italic font.

6.6.2.5.3

Change the table in subclause 6.6.2.5.3 of ISO/IEC 23008-12:2025/Amd 1:2025 to:

channel_id or packed_channel_id	Mapping (depending on the 'colr' box)		
0	Unused		
1	Unspecified		
2	Y	R	C
3	Cb	G	M
4	Cr	B	Y
5	Alpha		
6	Depth		
7	Reserved	Reserved	K
8-255	Reserved for future use.		

Add the following in subclause 6.6.2.5.3 of ISO/IEC 23008-12:2025/Amd 1:2025, after *channel_id*:

channel_id, when present, shall not be equal to 0 (Unused).

The regions indicated by all instances of channel_id and packed_channel_id not equal to 0 present in the current colour format enhancement derived image item, are used to generate a derived image based on the values of the indicated channel_id and packed_channel_id and any associated colour information.

If is_packed_flag is equal to 1, then the picture is first partitioned into (num_cols_minus1+1)×(num_rows_minus1+1) regions as follows:

The width and height of each region is computed as follows:

```
RegionWidthInLumaSamples = image_width / num_cols_minus1

RegionHeightInLumaSamples = image_height / num_rows_minus1
```

A region R[j][k] for any value of j in the range of 0 to num_rows_minus1, inclusive, and k in the range of 0 to num_cols_minus1, inclusive, is associated with the luma area starting from the horizontal and vertical coordinates OrigX[j][k] and OrigY[j][k] that are computed as follows:

```
OrigX[j][k] = (k×RegionWidthInLumaSamples)+(hor_guard_band_mul2×2)

OrigY[j][k] = (j×RegionHeightInLumaSamples)+(ver_guard_band_mul2×2)
```

and end at horizontal and vertical coordinates EndX[j][k] and EndY[j][k] that are computed as follows:

```
EndX[j][k]= ((k+1)×RegionWidthInLumaSamples)-(hor_guard_band_mul2×2+1)

EndY[j][k]= ((j+1)×RegionHeightInLumaSamples)-(ver_guard_band_mul2×2+1)
```

Only the luma samples associated with any indicated regions, as specified above, are considered for the generation of a derived image and all other luma and chroma samples are ignored.

6.8.12.1

Change the following sentence from subclause 6.8.12.1 of ISO/IEC 23008-12:2025/Amd 1:2025:

When the flag tile_info_present_flag is not set, the tile information of a layer of the image pyramid is derived depending on the image item as described in Tables 2 to 4 below.

To:

When the flag `tile_info_present_flag` is not set, the tile information of a layer of the image pyramid is derived depending on the image item as described in Tables 2 to 5 below.

Add the following Table 15 after Table 4 in subclause 6.8.12.1 of ISO/IEC 23008-12:2025/Amd 1:2025:

Table 15 — Tile information based on Tiled image item `'tili'`

`ImagePyramidEntityGroup` tile information	**Tiled image item `'tili'`**
tileWidth	`tile_width`
tileHeight	`tile_height`
tileColumns	ceil(*ispe*.`image_width/tile_width`)
tileRows	ceil(*ispe*.`image_height/tile_height`)

6.11

Add the following new subclause after subclause 6.10.4.1.3:

6.11 Tiled image items

6.11.1 General

A tiled image item is constructed of uniform, independently coded tiles arranged in rows, columns, and optionally extra dimensions, to form a rectangular image or n-dimensional hyperrectangle. The tiles are identical in size, format, coding, and makeup and may be compressed or uncompressed.

6.11.2 Tiled image item

An image item of type `'tili'` is a tiled image, with each tile coded independently from other tiles. Input tiles may be stored either in separate external files or in a contiguous range of addressing space to support byte range addressing and retrieval of individual tiles with a single read.

NOTE 1 The image coding method is defined by a writer using a valid image codec 4CC.

NOTE 2 As opposed to a `'grid'` image item, where the declaration and addressing of tiles occurs in the file-scoped MetaBox, a `'tili'` image item with contiguous range of addressing space has a single declaration parameter in the file-scoped MetaBox and an associated addressing table, with offsets and extents for each tile, stored with the image tiles, typically in a media data box. This has the advantage that the required file ranges of the addressing table, which can be large for terapixel images, can also be loaded on-demand.

The tiled image item (`'tili'`) shall be associated with `TiledImageConfigurationProperty` (`'tilC'`) and `ImageSpatialExtentsProperty` which carries the width and height of the overall tiled image.

NOTE 3 The tiled image item (`'tili'`) allows for storing YUV imagery as a single coded tile object. Conversely, each band in a tile can be stored separately, allowing for direct access to each component in a tile separately. This results in a 3D arrangement of tiles within the image. This is useful for multi and hyperspectral imagery.

NOTE 4 A tiled image item (`'tili'`) can be included within an `ImagePyramidEntityGroup`. To create a full pyramid, progressively binned tiled image items may be included. When building a multi-resolution pyramid, different image coding methods can be used for each layer.

The location of input tiles to the tiled image item is identified by the corresponding `DataEntryTiledItemBox` in the `DataReferenceBox` which is mapped to the tiled image item through `data_reference_index` in the `ItemLocationBox`.

The `DataEntryTiledItemBox` shall contain URLs to the input tiles stored in separate external files or shall contain the size and offset to the `TiledImageOffsetTable` to support byte range addressing and retrieval of individual tiles.

If the input tiles are stored in separate external files, then each external file shall contain only one input tile as an image item and be a HEIF compliant file. The external files shall not contain any entity grouping. The input tile in an external file may contain item properties associated with them. The `handler_type` of the tiled image item shall be equal to the `handler_type` of the input tile in the external file.

When `DataEntryTiledItemBox.external_tiles_urls` is equal to 0, all the necessary item properties for the input tiles of the tiled image item shall be present in the `ItemPropertyBox` and are associated to the tiles of the tiled image item through the `TileItemPropertyAssociationBox`. When `DataEntryTiledItemBox.external_tiles_urls` is different from 0, all the necessary item properties for an input tile stored in an external file are present in the external file.

When the overall image dimensions are not an even multiple of the image tile size, the rows are padded on the right to complete the last tile in each row of tiles, and the columns are padded on the bottom to complete the last tile in each column of tiles. The width and height parameters in the `ImageSpatialExtentsProperty` are set to the size of the image containing valid image content, effectively achieving a crop of the padded boundary area.

When the input tiles are stored in separate external files, the coding format of the tiled image item is given by the `item_type` of the image item in the external file. When the input tiles are stored in the same file as the tiled image item, the coding format of the tiled image item is given by `tile_item_type` in the `TiledImageConfigurationProperty` associated with the tiled image item.

6.11.3 Tiled image configuration property

6.11.3.1 Definition

Box type: `'tilC'`

Property type: Descriptive item property

Container: `ItemPropertyContainerBox`

Mandatory (per item): Yes, for image items of type `'tili'`

Quantity (per item): One

The `TiledImageConfigurationProperty` specifies parameters associated with a tiled image item (`'tili'`) (see Section 6.12). These parameters include the tile resolution, and the image item type used to code and store individual tile content. Configuration information also includes the number and size of additional dimensions when coding n-dimensional hyperrectangles. This includes support for the coding of multi and hyperspectral imagery where each band in a `tili` tile region is separately retrievable.

The `TiledImageConfigurationProperty` consists of `TileItemPropertyAssociationBox` which associates tiles to item properties in the `ItemPropertiesBox`.

6.11.3.2 Syntax

```
aligned(8) class TileItemPropertyAssociationBox
     extends FullBox('tipa', version=0, flags=0) {
     unsigned int(8) association_count;
     for (j=0; j<association_count; j++) {
          bit(1) essential;
          if (flags & 1)
               unsigned int(15) property_index;
          else
               unsigned int(7) property_index;
      }
}


aligned(8) class TiledImageConfigurationProperty
extends ItemFullProperty('tilC', version=0, flags=0) {
   unsigned int(32) tile_width;
```

```
    unsigned int(32) tile_height;
    unsigned int(8) number_of_extra_dimensions;
    for (int i=0; i<number_of_extra_dimensions; i++) {
       unsigned int(32) dimension_size[i];
       }
    if (DataEntryTiledItemBox.external_tiles_urls==0) {
       unsigned int(32) tile_item_type;
       TileItemPropertyAssociationBox tile_association;
    }
}
```

6.11.3.3 Semantics

`tile_width`, `tile_height` shall be set to the size of a single tile width and height. All tiles have the same size. Tiles at the right or bottom border of the overall image may include padding when the tile width and or height are not integer multiples of the overall tiled image item width or height. In this case, the `ImageSpatialExtentsProperty` is set to the boundary of the true image width and height to achieve a crop of the padded area.

`tile_item_type` specifies the image item type used for all the individual tile images. In a tiled image item, each tile is coded separately so it can be extracted and decoded independently. `tile_item_type` shall be set to a valid four-character code for a coded image item (e.g., `'hvc1'` for h265 compression, `'j2k1'` for JPEG2000, or `'unci'` for uncompressed). When required by the image item type, all necessary image properties shall be associated using the `TileItemPropertyAssociationBox`. Certain codecs (jpg, etc.) may not require any configuration properties in such a case the `association_count` in the `TileItemPropertyAssociationBox` is set to 0.

`number_of_extra_dimensions` specifies the number of extra dimensions if the image resembles a `(number_of_extra_dimensions+2)`-dimensional hyperrectangle. For a 2D image, `number_of_extra_dimensions` shall be 0.

For a single layer 2D image (e.g., YUV or monochrome), tiles are indexed with a uniform ordering of the form [y][x], where [x] is the width parameter, and [y] is the height parameter. For a multi-dimensional image where components are stored separately, the tiles are indexed with a uniform ordering based on the `number_of_extra_dimensions`, such as when equal to 0, the form is [y][x], when equal to 1, the form is [z][y][x], when equal to 2, the form is [a][z][y][x], and so on.

`dimension_size[i]` specifies the size of dimension i+2 of the n-dimensional hyperrectangle.

NOTE The size of the first two dimensions are the `image_width` and `image_height` specified in the `ImageSpatialExtentsProperty` of the `'tili'` item.

`association_count` indicates the number of item properties associated with the tiles of the tiled image item.

`essential` when set to 1 indicates that the associated property is essential to the tiles of the tiled image item, otherwise it is non-essential.

`property_index` is either 0 indicating that no property is associated (the essential indicator shall also be 0), or is the 1-based index (counting all boxes, including `FreeSpace` boxes) of the associated property box in the `ItemPropertyContainerBox` contained in the `ItemPropertiesBox`. `property_index` shall not be greater than the number of boxes contained in the `ItemPropertyContainerBox`. The (flags & 1) value in TileItemPropertyAssociationBox should be equal to 0 unless there are more than 127 properties in the ItemPropertyContainerBox.

6.11.4 Tiled image item data

The payload of a tiled image item (`'tili'`) consists of tiles of the item when the `external_tiles_urls` in the associated `DataEntryTiledItemBox` is set to 0.

6.11.5 Data entry tiled item box

6.11.5.1 Definition